AMENDMENTS TO THE CLAIMS

Sub D1

C1

1.(Amended Twice) A method of executing a software application, comprising the steps of:

(a) calling the software application residing on a server from ~~a client,~~ one of a plurality of clients, the ~~client~~ clients and the server connected to each other through ~~a~~ at least one network, the software application having a plurality of policy frameworks, each associated with a respective one of ~~a~~ the plurality of clients;

(b) launching a container/desktop of one of the ~~client~~ plurality of clients consistent with the respective policy framework of the one client;

(c) the container/desktop initializing and communicating to the server to execute a script of the application;

(d) executing the script on the server, the script downloading a first user-interface component of the application to the container/desktop;

(e) the container/desktop executing the first user-interface component;

(f) the first user interface component linking to and starting a subsequent user-interface component of the script,

(g) closing the first user-interface component; and

(h) the server downloading the subsequent user-interface component to the container/desktop, and the container/desktop executing the subsequent user- interface component and then closing the subsequent user-interface component.

2. (Previously Cancelled).

3. (Previously Amended) The method of claim 1, further comprising the container/desktop removing the user-interface components from memory within the client as each first and subsequent user-interface component is closed.

C1

4. (Amended Twice) A method of executing an application having a plurality of tasks to be interactively executed with a user, said method comprising:

(a) downloading to ~~a container/desktop~~ one of a plurality of container/desktops only those of a plurality of user-interface components consistent with a policy/framework of the one container/desktop, the user interface components stored on a server needed to perform a first task of the plurality of tasks of an application according to a script executing on the server;

(b) executing the first task on the container/desktop;

(c) closing said downloaded user-interface components needed to perform the task when no longer needed;

(d) purging said closed user-interface components from said one container/desktop when said closed user-interface ~~component is~~ components are no longer needed;

(e) downloading to ~~a~~ the one container/desktop only those user-interface components stored on a server needed to perform a subsequent task of the plurality of tasks of an application according to the script;

(f) executing the subsequent task on the one container/desktop;

(g) repeating steps (c) and (d); and

(h) repeating steps (e) through (g) until all of the plurality of tasks are completed.

5. (Currently Amended) A computer server, comprising:

(a) a processor, a memory, a bus, and at least one I/O port by which to communicate with a remote client having a container/desktop,

C1

(b) an operating system with which to coordinate the processor, the memory, the bus and the at least one I/O port to communicate to the client;

(c) an application comprising a plurality of tasks to be executed on the container/desktop, the application stored in memory of and executing on the server;

(d) a script of the application stored in the memory of and executing on the server; and

(e) a plurality of user-interface components stored in the memory, the script comprising code executing on the server to connect the user-interface components to comprise the application;

wherein the application launches the container/desktop on the client ~~which container/desktop~~ that interacts with the script executing on the server to download from the server to the container/desktop only those user-interface components required for a current task executing on the container/desktop.

6. (Currently Amended) A client device, comprising:

(a) a container/desktop;

(b) an I/O port with which to communicate to one or more servers having software applications invoking a plurality of tasks on the container/desktop, scripts, and user-interface components for the applications; and

(c) an interactive medium with which to interact with a user,

C1

wherein when the user uses the interactive medium to request an application from the server, the container/desktop communicates with the server through the I/O port and invokes a script of the application in the server, the script and the application executing on the server and which downloads only those user- interface components to the container/desktop needed by a current one of the plurality of tasks executing according to the script and ~~and~~ wherein the container/desktop discards the user-interface components no longer needed by the application.

7. (Original) The client device of claim 6 wherein the container/desktop comprises code for implementation of the user-interface component on a personal computer.

8. (Original) The client device of claim 6 wherein the container/desktop comprises code for implementation of the user-interface component on a voice-response unit.

9. (Original) The client device of claim 6 wherein the container/desktop comprises code for implementation of the user-interface component on a network computer.

10. (Original) The client device of claim 6 wherein the container/desktop comprises code for implementation of the user-interface component on a pervasive mobile device.

11. (Original) The client device of claim 6 wherein the container/desktop comprises code for implementation of the user-interface component on a second server behaving as a client.

12. (Currently Amended) A method to script user-interface components to create an application ~~which is~~ stored on a server and whose user-interface components are downloaded to one of a variety of container/desktops of different clients, said method comprising:

C1

(a) decomposing the presentation logic of the application into a plurality of tasks to be performed interactively with a user on the client;

(b) for each of the tasks, creating a state diagram having a plurality of nodes wherein a user-interface component is associated with at least two of the plurality of nodes;

(c) writing a script connecting each of the user-interface components in accordance with the state diagram and a policy framework of the container/desktop, each one policy framework being unique to one of said variety of container/desktops of different clients; wherein said script and said user-interface components are stored on at least one server to which said client is connected and said script executes on said server to download said user-interface components ~~are downloaded~~ to said container/desktop of said client on an as-needed basis and in accordance with the policy framework unique to said container/desktop of said client.

13. (Currently Amended) A program product that causes an application to be executed on a server by a user of a client device in which the user interacts through a container/desktop of the client device with the application, the program product comprising a script of a plurality of user-interface components, such that the container/desktop initializes the script, the script executing on the server and said container/desktop ~~and controls~~ controlling the downloading and termination of the user-interface components on an as-needed basis according to the script~~, the script executing on the server~~.

14. (Original) The program product of claim 13, wherein the user-interface components are stored on one or more servers.

C1

15. (Original) The program product of claim 13 wherein some or all of the user-interface components are stored on the program product.

16. (Currently Amended) A computing apparatus, comprising:

(a) ~~means~~ a server for storing a software application to be performed by a user on a container/desktop;

(b) means for invoking a script of a plurality of user-interface components to execute said software application having a plurality of tasks according to a policy framework of said container/desktop;

(c) means for executing said script on the server to perform the plurality of tasks of said software application;

(d) means for downloading only those of said plurality of user-interface components from the server to said container/desktop when a corresponding task is being accomplished according to said script; and

(e) means to display each of said user-interface components and to discard any previous of said user-interface components from said container/desktop when no longer needed by said user to perform said corresponding tasks of said software application in accordance with said policy framework of said container/desktop.